Bruce Xi

(236)-777-8218 • brucexi99@outlook.com • bruce-shidi-xi • brucexi999

Education

The University of British Columbia **Vancouver, BC**

Master of Engineering in Electrical and Computer Engineering, GPA 4.0 *2021–Present*

- Relevant courses: Digital/Microcomputer System Design, Computer Architecture, VLSI, High-speed Data Link, IC Testing and Reliability, DNN Accelerator, Deep Learning

Imperial College London **London, UK**

Bachelor of Engineering in Materials Science and Engineering, First-Class Honours Degree *2018–2021*

- Obtained Dean's List for three consecutive years (2018-2021)

Project

Design of a 10 Gbps Differential SerDes Data Link with TX FIR Equalization **Jan.-May 2024**

- Designed a 10 Gbps differential SerDes data link using Cadence Virtuoso in 45 nm technology, incorporating a TX, a differential channel, and an RX. Performed circuit simulations using Spectre, achieving a power consumption of 9 mW, an eye-opening of 47 mV under the worst-case data pattern, and zero errors with both worst-case and PRBS7 input over 10,000 UIs.
- Characterized channel impulse and pulse responses using MATLAB, determining the worst-case data pattern and computing tap values for 2-tap FIR equalization.
- Developed the TX with a 4:1 serializer based on a 5-latch-2:1 multiplexer topology with half-rate clocking, requiring 5 GHz clock frequency at max. Pre-drivers are designed as inverter chains optimized using logical effort. Implemented a high-swing, voltage-mode driver segmented into 10 parts to provide a tunable impedance of $50\Omega \pm 30\%$ and configured to pre-emphasize the signal, functioning as an FIR equalizer to suppress ISI.
- Implemented the RX with 1:4 deserialization at 2.5 GHz using four slicers followed by synchronizers. Each slicer integrates a track-and-hold switch executed as a transmission gate, a StrongArm latch, and an SR latch.

AMBA AXI Stream Header Insertion **Feb.-Mar. 2024**

- Designed an RTL module using Verilog to insert headers into network data packets. This module transfers data following the AXI Stream protocol, accepting data and headers from two master interfaces. It removes invalid bytes from both input signals based on the keep signal (packing), merges the remaining valid bytes, sends them to a slave interface, and outputs the corresponding keep and last signals.
- Used pipelining for data and handshake signals, and equipped skid buffers to handle backpressure, eliminate bubbles in the pipeline, and optimize throughput.
- Verified the module using SystemVerilog random stimuli to ensure data correctness with no losses or duplications.

Concurrent VLSI Routing with Multi-agent Deep Reinforcement Learning **May-Oct. 2023**

- Developed a Python-based machine learning model to address the VLSI global routing problem in a concurrent manner. Modeled routing as a pathfinding task and solved it using multi-agent reinforcement learning integrated with deep neural networks (MLP and GNN implemented in PyTorch).
- The proposed work overcame the traditional net-ordering issue, guaranteed zero overflow, and outperformed an A* baseline by 2.6% in terms of wirelength.

Microcomputer System Development **Jan.-Apr. 2023**

- Developed critical components for a microcomputer system capable of operating at 45 MHz on an Altera FPGA. Implemented an 8-way set associative cache using SRAM on the FPGA, and designed a segment of the cache controller featuring a Tree PLRU (Pseudo Least Recently Used) replacement policy using Verilog. Also designed components of a DRAM controller using Verilog.
- Wrote embedded C drivers to manage a soft Motorola 68000 microprocessor for communications via SPI protocol with a Flash, and via IIC protocol with a EEPROM and an ADC/DAC PCB.
- Leveraged hardware timer interrupts to create a snake game in C, which was playable on the system and utilized VGA for user interaction.

Work Experience

Motorola Solutions **Vancouver, BC**

Design Validation Co-op *May-Dec. 2022*

- Conducted extensive electrical, mechanical, and optical tests on surveillance cameras.
- Proposed reflections and innovations on existing testing processes, self-taught Python programming, and wrote software for test automation and data analysis. Reduced repetitive manual labor within the team, improving work efficiency. Automation of some tests reached up to 90%.
- Collaborated within a team using Git, GitHub, Confluence, and Jira to optimize workflow efficiency.

Skills

Hardware: Verilog, SystemVerilog, FPGA, ModelSim, Quartus, Cadence Virtuoso

Software: Python > C > ARM Assembly = C++, Linux, MATLAB